International Bar Code

STAx Series
User Guide

01 AUG 17

Contents

Version 01.08 New Features

Firmware Updating

STAx readers can be flashed with new firmware by using IBC's Flash Utility.

Proximity Card Reading

STAx readers ordered with proximity input type "H" can read and process the following proximity cards: IBC, Hid, EM, Casi, Awid, Farpointe. For best read range, we recommend using IBC formatted cards.

Background Downloading

STAx readers support full downloading of data into an alternate memory bank without affecting operation of the reader.

Differences between STA V7 readers and STAx readers

Enhanced Barcode Reading

The barcode algorithms in the STAx series readers are more agressive than those in the STA series, allowing you to read difficult-to-read barcodes. STAx series readers can also read barcodes which are in close proximity to other writing on cards.

Magnetic Stripe Reading

STAx readers support track 1 magnetic stripe data or track 2 magnetic stripe data. Track 3 is not supported at this time. Track 1 and Track 2 data cannot be concatenated to form one data item.

Multiple Memory Banks

STAx supports 2 memory banks. The reader can validate card reads against one bank of memory, while the reader is being downloaded with new data into the other memory bank. Full downloads into the reader can be achieved without the need to take the reader offline while downloading.

Barcode Reading

STAx readers do not support the reading of code 93.

Schedules

STAx schedules cannot cross the midnight boundary.

Mulidrop(protocol) Mode

STAx readers do not support protocol mode.

Proximity Card Reading

STAx readers contain updated electronics enabling reading of the following types of cards: IBC, Hid, EM, Casi, Awid, and Farpointe. Indala cards are no longer supported.

Enrolling at Reader

STAx readers doe not support the enrolling or deletion of cards at the reader. All additions and deletions must be made through software.

Upload Commands

The STA reader contained an upload command "u" which could be used to upload ID records stored in the reader. STAx readers do not support this feature.

Circular Log Buffer

The STAx reader contains a circular buffer for log data, therefore the cl command is no longer needed. It is however still supported.

Commands

Command for stand alone functionality (downloading ID's, etc) are the same as older STA readers, however commands which are used to program symbologies and other parameters have changed. Those commands are similar to the newer "JX" style commands. STAx readers cannot be programmed using barcodes or magstripes.

Flash Memory

The STAx reader uses flash style memory, therefore data in the reader cannot be modified without clearing it first.

For example, STA readers allowed you to change a particular schedule without first deleting it. That cannot be done with the STAx reader. All schedules need to be cleared first, then the updated schedule information can be downloaded.

The same applies to access ID numbers. If an ID number is deleted, that space is not re-claimed. This should not be an issue since the STAx readers have more memory than older STA readers, and ID number changes are easily accomplished by using the background down-loading techniques.

There is no background downloading capability for schedules, so schedules need to be cleared first prior to downloading. Once downloaded, additional (unused) schedules can be added at any time.

Software developers should take this into consideration when designing or modifying their code to support the STAx readers.

It also takes more time to clear (erase) flash memory. Older STA readers used ram rather than flash, which was very quick to clear. Software developers should not assume that whenever they send a command to reset the reader, or clear all the IDs, etc... that the command will finish immediately. Some commands can take up to 20 to 30 seconds to execute.

Add Employee ID Command

The STA reader had two commands to add an employee id number into the STA. These commands were AN.... and AC.... The AN command did not check for duplicate ID numbers and simply added the ID into the list. The AC command would first check to see if the ID was i the list and return an error if it was found. The STAx reader does not check for duplicates. The AC command is supported for compatibility purposes, but it acts exactly like the AN command.

Modifying Current Software to support STAx readers.

Some STA V7 commands do not have a natural response. Software developers would simply assume that the command had been executed.

Those commands are:

/xx	reset record size and clear memory
|x	set mode
"ss	set relay/green led time
`ss	set bad ID led blinking
:x	bad id logging
cl	clear log
cd	clear downloaded list

The same assumptions could be used when writing software to control STAx readers except that those commands which modify memory (in this case flash memory) will take longer to complete. The Reset Recordsize command could take up to 40 seconds.

STAx readers feature an option named "full response" to handle this scenario. If "full response" is turned on, then the reader will always respond with "OK" or "NOK" to every command which does not have a natural response.

Software developers can modify their software to first send a version command to the reader to see if the reader is a STA reader or STAx reader. If it is an STAx reader, then they can look for the OK or NOK response from the reader in their code. This is the suggested way to modify code. IBC can insert custom version responses for those VARS that request it.

Alternatively, since the first 5 command listed above are typically done only once before installation, they can be sent manually using one of IBCs terminal programs, followed by a V command immediately. The V command will be queued by the reader and when the response is seen by software, it will know that the prior command had completed.

The Stax reader now allows ID data to be downloaded in the background, into a separate memory bank, while the reader is operational. Developers may want to take advantage of this option. The method of download would be as follows:

1) Send BACK command to tell the reader you want to download to the background
2) Send the cd command to clear the downloaded list and wait for the OK
3) Download records exactly as you would with the STA reader
4) When done, send the SWAP command which "swaps" the memory banks. The latest data which you have downloaded becomes the active data for the reader.

Note that there is no facility for downloading schedule information in the background. The same schedule information is used for both banks of memory.

If any schedule needs to be changed, the only way to do that is to send the new SC (schedule clear) command which will clear the schedules, and then you can download the new schedule information. This command however is very quick and should have minimal impact on users.

Functionality

The Stax readers are stand-alone access control readers and do not require the use of an access control controller. The STAx readers are in fact a reader and controller combined into a small reader form factor.

Major features of the reader are:

- 2MB of flash memory for storing access control data and usage logs.
- Automatic relay triggering on acceptable reads.
- Time and Date stamping for all transactions.
- Optional Time Display.
- RS232 or TCP communications for downloading data and uploading logs.
- Easily programmed using serial commands.
- Barcode and magstripe masking.
- Logging of invalid cards.
- Red and Green status leds.
- "JX" series reader emulation.

The STAx reader contains 2MB of memory which contains an Access Control list of valid Employee ID numbers. When a card is scanned or swiped, and the number on that card matches an Employee ID number in the Access Control list, the internal relay is triggered to open a door.

In addition to containing the Access Control list, each employee in the list may also be restricted to entering only during specific times (or days). This is achieved by assigning the employee to a specific schedule.

The STAx reader, in addition to maintaining the access control list, also saves each transaction in memory, so that the transactional information can be used later for time and attendance.

The STAx readers can operate in "JX" reader emulation mode. In this mode, the reader will operate like a standard IBC "JX" series reader, utilizing the same command sets and

functionality as a standard "JX" reader. This feature allows the STAx reader to be utilized either as a stand-alone reader or an on-line reader.

Access Control

The STAx reader provides stand-alone access control for one door, utilizing a list of valid employee id numbers and allowable entrance times and days.

When a valid ID card is scanned through the reader (a card matching an ID in the Access Control list), the internal relay is triggered and the door is opened.

The Access Control List is downloaded from a computer.

The maximum length of an ID stored in the reader Access Control list is 40 characters, and the minimum is 4 characters.

If you need to read cards which are longer than 40 characters, then you can mask the information on the card to make an ID which is 40 characters.

In addition to validating the ID number, the reader also will validate the ID against an internal table of schedules which can be loaded into the reader. Up to 100 different schedules can be loaded into the reader. Schedules are added by downloading from a computer.

Each schedule contains a start and finish time for all seven days of the week. If an ID in the list is associated to a schedule, then not only must the id match, but also the time and day must be within the boundaries of the schedule.

Relay

The relay contained in the reader is a form C relay (one common contact, one normally open contact, and one normally closed contact). This relay is rated at 0.5A @ 30VDC. Do not attempt to pull more than 0.5A@30V of current through the relay or the relay may become blown. If you need to draw more current than our relay will support, then we recommend using a power relay, and triggering the power relay from our internal relay. The relay is protected internally for backward voltage spikes by a MOV (metal oxide varistor). We also highly recommend that you install a MOV across the solenoid leads of your electric strike (or whatever door mechanism you are using) to avoid voltage spikes on the line causing a problem with the reader.

Wiring

All STAx readers come with 2 sets of wires - one wire for the communications and power and another wire for the relay control.

The RS232 communication wire is usually a black wire containing the following:

Red	+12VDC or +24VDC
Blue	Ground
Green	Reader Transmit
Yellow	Reader Receive

The TCP communications wire is usually a 3' CAT5 cable with a RJ45 network plug.

The relay wire is a black wire (flat telephone style wire) containing the following:

Red	Common
Green	Normally Open
Yellow	Normally Closed

If your reader was supplied from IBC with a DB9 or DB25 connector, then the DB connector can be connected directly to a PC for communications. Usually, units supplied with a DB9 or DB25 connector are also supplied with an AC adaptor (110V 60hz) for power. IBC supplies either a standard american 110/60 or european 220/50 ac adaptor. We do not supply power supplies for other voltage systems.

If you are not using the ac adaptor, then you must provide power to the reader using the appropriate voltage. If you reader model ends in a 2, then it is a 12VDC reader which can be powered with any DC voltage from 8VDC to 15VDC. If the reader model ends in a 4, then you can power it with any voltage between 15VDC and 30VDC.

Leds

SA/STA readers contain two leds on the front of the reader - one red and one green.

These leds have special meaning depending on the mode of operation the reader is in.

While in the "JX" emulation mode, both leds are normally off, and operate according to the "JX" programming defaults which you have programmed the reader to.

While in the STAx mode, the red led is normally on, and the green led is normally off. Whenever a card is scanned or swiped which matches a number in the Access Control list (access granted), the green led will blink. If a card is scanned or swiped and that card is not loaded in the Access Control list (or doesn't match the scheduling requirements), the red led will blink.

While in SA/STA mode, you cannot program the leds to do anything else other than the functions described above.

Schedules

You can load up to 100 different schedules into the reader. These schedules contain a start and end time for each day of the week for which an employee will have access.

For example, if you have an employee which can enter only between the hours of 7AM and 3 PM on Mondays, you would set up a schedule which contains 0700 thru 1500 for Mondays, and then associate that employee with this particular schedule.

By default, there are no preloaded schedules loaded into the system. You therefore must load each schedule by downloading into the reader using the serial port.

Each employee which is entered into the reader must have a schedule # associated with it. If you wish to allow an employee to have access at all times, then you can use the schedule number of 0, which is a default schedule # set up to allow access at all times.

When schedules are loaded into the reader, you must enter in a start and end time for each of the seven days of the week. It is ok to cross midnight (i.e. starting Monday at 2000 and ending Tuesday at 0600).

Employees which are entered into the system by hand (by scanning in cards), are automatically assigned the schedule number of 0. There is no way to assign schedule numbers to cards which are entered in by hand, other than sending a serial command.

Schedules may be associated together so that 2 or more schedules are used for authorization in tandem when validating the schedule for an employee. See the Schedule section of the programming commands for more information on this.

Time Display

The STAx series readers may be ordered with an 8-character alphanumeric display. This display will automatically show the time which is set in the reader and will change every minute. All scans which are stored in the reader will have a time stamp which matches the clock at the time of the scan.

Barcode Reading

STAx series readers can read the following barcode symbologies:

- Code39
- Interleaved 2 of 5
- Standard 2 of 5
- Codabar
- Industrial 2 of 5
- Ean 13
- Upc A
- Code 128

Read Direction

Barcodes can be read in either direction. A direction indicator ("f" for forward, "r" for reverse) can be pre-pended to the barcode data to show the direction the card was scanned.

Masking

Masking (discussed later in the manual) can be used to select only portions of the barcode to process, and also add characters to the data when being processed. There aer two possible masks which can be turned on. Each mask operates based on a barcode of a specific length.

Check Digits

The reader can optionally check and verify check digits which may be included in the barcode for those symbologies that allow check digits. Note that Upca and Ean13 inherently use check digits and cannot be turned off. Check digit verification and transmission can be turned on or off for all of the other symbologies.

Prefix and Suffix Characters

Up to 5 prefix characters and 5 suffix characters can be added to the barcode data. Prefix characters are especially useful to identify the data source on dual and tri-technology readers, when using readers with a serial output.

Length Restrictions

The reader can be programmed to ignore barcodes which are not a specific length. You can enter in a different length requirement for each symbology, except Upca and Ean13.

Turning Symbologies on and off

All symbologies are turned on by default. Each symbology can be turned on or off by command.

Codabar Clsi Conversion

Codabar Clsi conversion can be turned on or off.

Usid/Cac Card Processing

STAx readers can optionally process 18, 21, and 24 character Usid cards, and 18 character Cac cards. For both cards, either the Pdi or Edi is extracted and converted to a base 10 number. The card instance identifier (on 18 character cards only) can also be appended to the data.

For Usid 21 and 24 character cards, you can elect to process (or not process) active cards, retired cards, reserve cards, sponsor cards, dependent cards, civilian cards, and reserve retired cards. The datatype taken from the 21 character card is the Pdi. For dependent cards, you can elect to take the dependent's Pdi, or the spondor's Pdi.

For 18 character Usid and Cac cards, there are only two card type options - to include dependent cards, and include all other cards. The datatype for dependents can be the dependent's Edi, or the sponsor's Edi. The datatype for all other cards can be the Pdi, or Edi. You can also choose to append the card instance identifier to the output data.

Kronos Card Conversion

STAx readers can autodiscriminate Kronos card formats if the Kronos Mode option is on. If on, then 17 digit code39 cards and 18 digit I2/5 cards are assumed to be in the Kronos format. There are two options - mode 1, using a 5 digit ID number, and mode 2, using a 9 digit ID number. The ID number is the 5 or 9 digits to the left of the facility code. For both modes, the site code is set to 0.

Alpha Delete

One of the general barcode options is Alpha Delete. If on, then any alpha characters in the barcode are ignored. This is useful when reading a barcode that has an alpha, but the desired output is wiegand (wiegand cannot have non-numeric characters).

Read Normalization

If there are two barcodes on the card in the same read path, or there is writing on the read path, then you can specify which one is the actual barcode to read. Yoru specify the "smaller" or "larger" one - smaller or larger referring to the number of black "bars" the reader will see.

Good Read Beep

You can turn on a good read beep to occur after reading a valid barcode. If you read a barcode which does not match length specifications, is not valid for a programmed mask, or cannot be decoded into a wiegand format because the data size is too large, then the reader will not beep.

Timeout

The read timeout is programmable, from 1 to 9. The timeout is a relative amount of time after seeign the last black bar, that the reader then begins to decode the barcode data. The default timeout is 5. In some cases, adjusting the timeout can help to read barcodes on cards that have unusually wide white spaces, or writing very close to the barcode.

Proximity Reading

STAx series readers can read the following proximity cards:

- IBC (all formats)
- Hid
- Awid 26 bit
- Casi Proxlite (12 digit mode)
- EM
- Farpointe

Proximity Format

Up to 3 different card formats can be programmed into the reader. The programmed formats are irrespective of card types and are fully user programable.

Turning Card Types on and off

All card types are turned on by default. Each card type can be turned on or off by command.

Programming Commands

The STAx series support two different types of commands:

Standard JX programming commands which start with an X, followed by a 3-digit command.
Standard STAx programming commands which are compatible with STA V7 series software.

Programing the reader can only be done by sending the commands serially.

All commands must end with a carriage-return (hex 0d).

Note that command parameter values are loosely checked by the reader. Please verify the validity of all commands prior to entering them into the reader. Not all commands are verified.

The command descriptions in this manual contain spaces but these spaces are for readability only. Do not put the spaces in the commands.

Barcode Programming Commands

Barcode programming consists of a number of commands. There is one General Setup command specifying global options for barcode processing. Each barcode symbology also has a general options command specific to that symbology, as well as commands to program prefixes and suffixes for each symbology. There are also a number of masking commands which can globally be used with barcode reading.

General Barcode Options Command

X037 t n a m k b c s

t	=	read timeout	1 thru 9, 5 is default
n	=	normalize option	0=off, 1=small, 2=large
a	=	alpha delete	0=disabled, 1=enabled
m	=	masking	0=disabled, 1=enabled
k	=	kronos mode	0=disabled, 1=mode 1, 2=mode 2
b	=	good read beep	0=disabled, 1=enabled
c	=	usic/cac option	0=disabled, 2=cac/usid only, 1=accept cac/usid and all others

Notes

- Read timeout "t" is an adjustable amount of time before the reader times out when it believes it has seen the end of a barcode. It is best to leave this timeout at 5. Reduce the value to have the reader timeout quicker.
- If multiple barcodes are contained left to right on a card, or there is writing next to a barcode, set the "n" parameter to choose which one to read.
- Set "a" to a 1 to remove all alphas found in the barcode. This is used to mask out alphas in cases where they are not needed or not allowed, such as in a wiegand output reader.
- Set "m" to a 1 to turn barcode masking on. Masking is either turned on or off for all symbologies. It is not possible to mask only specific symbologies.
- Set "k" to 1 to enabled Kronos mode 1, or set it to 2 to enable Kronos mode 2. Mode 1 creates a 2/5 string containing the rist 2 digits of the site code followed by the last 5 digits of the id number. Mode 2 creates a 0/0 string containing a site code of 0 and the last 9 digits of the id. Code 39 and I2/5 formats are auto discriminated.
- Set "b" to a 1 to turn on the good read beep. Note that the good read beep will not beep if a barcode has been read but cannot be transformed into the proper output value. For example, if you read a 5 digit barcode "99999" and attempt to transmit that as an id number within a standard 26 bit wiegand format you will not get a good read beep since the number 99999 is too large for the programmed format.
- The "c" option is used to program the reading of code39 barcodes on the cac card and code39 barcodes on the usid card. if "c" is 0 then cac/usid cards are treated as ordinary cards and no special processing occurs. if "c" is

a 2 then only cac/usid cards are read (18,21, or 24 character code39 cards). The card numbers are extracted and converted based on the "s" parameter. If "c" is a 1 then all cards are read normally except cards which contain cac/usid formats are automatically discriminated and processed based on the "s" parameter. Note that cac mode and kronos mode should not be turned on at the same time, as the two formats cannot always be autodiscriminated.

Code39 General Setup

X001 e c t s d ll x p s

e	= enable code 39	0=disabled, 1=enabled
c	= check digit	0=disabled, 1=enabled
t	= transmit check digit	0=disabled, 1=enabled
s	= transmit start/stop	0=disabled, 1=enabled
d	= transmit direction indicator	0=disabled, 1=enabled
ll	= length	restrict length of barcode to ll
x	= extra bars	# extra bars allowed
p	= prefix	0=disabled, 1=enabled
s	= suffix	0=disabled, 1=enabled

notes

- e parameter must be 1 in order to read code39 barcodes
- If p is set to a 1, then the programmed prefix is prepended to the barcode data.
- Note that the prefix can be up to 5 characters in length, and is prepended to the code39 barcode data after masking has taken place, if any.
- If c is set to a 1, then the code39 barcode you are reading must have a check digit in it in order to be valid. The check digit is not transmitted unless the t parameter is set to a 1.
- If s is set to a 1, then the code 39 start/stop characters (*) are sent along with the barcode data.
- If d is set to a 1, then a direction indicator is prepended to barcode data. The direction indicator is either an "f" for forward scan, or "r" for reverse scan.
- If s (suffix) is set to a 1, a suffix of up to 5 characters may be added to the end of the barcode.
- The ll parameter specifies the length of the barcode to read. If length is set to a non zero value, then the code39 barcode read must match that length or it will be ignored. The length does not include start and stop characters, or the check digit. Length checking occurs prior to kronos or cac/usid checking, so specific kronos or cac/usid cards can be ignored based on the length parameter.
- The x parameter refers to the number of allowable extra bars that the reader can use when attempting to decode a code39 barcode. It is suggested that this parameter be left at the default of 2.

Code39 Barcode Prefix

X016 aaa bbb ccc ddd eee

aaa= prefix character 1	3 digit ascii character, 000 denotes end
bbb= prefix character 2	3 digit ascii character, 000 denotes end
ccc= prefix character 3	3 digit ascii character, 000 denotes end
ddd= prefix character 4	3 digit ascii character, 000 denotes end
eee= prefix character 5	3 digit ascii character, 000 denotes end

notes

- Up to 5 prefix characters can be programmed for code39 barcodes. If the prefix option is turned on, then these prefix characters are prepended to the code39 barcode data.
- Prefix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065. A value of 000 means the end of the prefix. The reader will not process any prefix charcters after a 000. A null, therefore, cannot be entered in as a prefix character. The prefix 065 066 067 000 000 is valid, and will prepend the barcode with "ABC". The prefix sequence 065 066 000 067 000 will prepend only an "AB" because the 000 signifies the end.
- The largest decimal value (ascii char) that can be used as a prefix is 126.

Code39 Barcode Suffix

X017 aaa bbb ccc ddd eee

aaa= suffix character 1	3 digit ascii character, 000 denotes end
bbb= suffix character 2	3 digit ascii character, 000 denotes end
ccc= suffix character 3	3 digit ascii character, 000 denotes end
ddd= suffix character 4	3 digit ascii character, 000 denotes end
eee= suffix character 5	3 digit ascii character, 000 denotes end

notes

- Up to 5 suffix characters can be programmed for code39 barcodes. If the suffix option is turned on, then these suffix characters are appended to the code39 barcode data.
- Suffix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065. A value of 000 means the end of the suffix. The reader will not process

any suffix charcters after a 000. A null, therefore, cannot be entered in as a suffix character. The suffix 065 066 067 000 000 is valid, and will place "ABC" after the barcode. The suffix sequence 065 066 000 067 000 will place only an "AB" after the barcode because the 000 signifies the end.

- The largest decimal value (ascii char) that can be used as a suffix is 126.

Standard 2 of 5 General Setup

X002 e c t d ll x p s

e	=	enable 2of5	0=disabled, 1=enabled
c	=	check digit	0=disabled, 1=enabled
t	=	transmit check digit	0=disabled, 1=enabled
d	=	transmit direction indicator	0=disabled, 1=enabled
ll	=	length	restrict length of barcode to ll
x	=	extra bars	# extra bars allowed
p	=	prefix	0=disabled, 1=enabled
s	=	suffix	0=disabled, 1=enabled

notes

- e parameter must be 1 in order to read standard 2 of 5 barcodes
- If p is set to a 1, then the programmed prefix is prepended to the barcode data. Note that the prefix can be up to 5 characters in length, and is prepended to the 2 of 5 barcode data after masking has taken place, if any.
- If c is set to a 1, then the 2 of 5 barcode you are reading must have a check digit in it in order to be valid. The check digit is not transmitted unless the t parameter is set to a 1.
- If d is set to a 1, then a direction indicator is prepended to barcode data. The direction indicator is either an "f" for forward scan, or "r" for reverse scan.
- If s (suffix) is set to a 1, a suffix of up to 5 characters may be added to the end of the barcode.
- The ll parameter specifies the length of the barcode to read. If length is set to a non zero value, then the 2 of 5 barcode read must match that length or it will be ignored.
- The x parameter refers to the number of allowable extra bars that the reader can use when attempting to decode a 2 of 5 barcode. It is suggested that this parameter be left at the default of 2.

Standard 2 of 5 Barcode Prefix

X020 aaa bbb ccc ddd eee

aaa= prefix character 1	3 digit ascii character, 000 denotes end
bbb= prefix character 2	3 digit ascii character, 000 denotes end
ccc= prefix character 3	3 digit ascii character, 000 denotes end
ddd= prefix character 4	3 digit ascii character, 000 denotes end
eee= prefix character 5	3 digit ascii character, 000 denotes end

notes

- Up to 5 prefix characters can be programmed for 2 of 5 barcodes. If the prefix option is turned on, then these prefix characters are prepended to the 2 of 5 barcode data.
- Prefix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065. A value of 000 means the end of the prefix. The reader will not process any prefix charcters after a 000. A null, therefore, cannot be entered in as a prefix character. The prefix 065 066 067 000 000 is valid, and will prepend the barcode with "ABC". The prefix sequence 065 066 000 067 000 will prepend only an "AB" because the 000 signifies the end.
- The largest decimal value (ascii char) that can be used as a prefix is 126.

Standard 2 of 5 Barcode Suffix

X021 aaa bbb ccc ddd eee

aaa= suffix character 1	3 digit ascii character, 000 denotes end
bbb= suffix character 2	3 digit ascii character, 000 denotes end
ccc= suffix character 3	3 digit ascii character, 000 denotes end
ddd= suffix character 4	3 digit ascii character, 000 denotes end
eee= suffix character 5	3 digit ascii character, 000 denotes end

notes

- Up to 5 suffix characters can be programmed for 2 of 5 barcodes. If the suffix option is turned on, then these suffix characters are appended to the 2 of 5 barcode data.
- Suffix characters are entered in as a 3 digit ascii number, representing thedecimal number of the ascii code. For example, "A" is a hex 41, decimal 065.
- A value of 000 means the end of the suffix. The reader will not process any suffix charcters after a 000. A null,

therefore, cannot be entered in as a suffix character. The suffix 065 066 067 000 000 is valid, and will place "ABC" after the barcode. The suffix sequence 065 066 000 067 000 will place only an "AB" after the barcode because the 000 signifies the end.

- The largest decimal value (ascii char) that can be used as a suffix is 126.

Industrial 2 of 5 General Setup

X003 e c t d ll x p s

e	=	enable ind 2of5	0=disabled, 1=enabled
c	=	check digit	0=disabled, 1=enabled
t	=	transmit check digit	0=disabled, 1=enabled
d	=	transmit direction indicator	0=disabled, 1=enabled
ll	=	length	restrict length of barcode to ll
x	=	extra bars	# extra bars allowed
p	=	prefix	0=disabled, 1=enabled
s	=	suffix	0=disabled, 1=enabled

notes

- e parameter must be 1 in order to read industrial 2 of 5 barcodes
- If p is set to a 1, then the programmed prefix is prepended to the barcode data. Note that the prefix can be up to 5 characters in length, and is prepended to the ind 2 of 5 barcode data after masking has taken place, if any.
- If c is set to a 1, then the ind 2 of 5 barcode you are reading must have a check digit in it in order to be valid. The check digit is not transmitted unless the t parameter is set to a 1.
- If d is set to a 1, then a direction indicator is prepended to barcode data. The direction indicator is either an "f" for forward scan, or "r" for reverse scan.
- If s (suffix) is set to a 1, a suffix of up to 5 characters may be added to the end of the barcode.
- The ll parameter specifies the length of the barcode to read. If length is set to a non zero value, then the ind 2 of 5 barcode read must match that length or it will be ignored.
- The x parameter refers to the number of allowable extra bars that the reader can use when attempting to decode a ind2of5 barcode. It is suggested that this parameter be left at the default of 2.

Industrial 2 of 5 Barcode Prefix

X030 aaa bbb ccc ddd eee

aaa= prefix character 1	3 digit ascii character, 000 denotes end
bbb= prefix character 2	3 digit ascii character, 000 denotes end
ccc= prefix character 3	3 digit ascii character, 000 denotes end
ddd= prefix character 4	3 digit ascii character, 000 denotes end
eee= prefix character 5	3 digit ascii character, 000 denotes end

notes

- Up to 5 prefix characters can be programmed for ind2of5 barcodes. If the prefix option is turned on, then these prefix characters are prepended to the ind2of5 barcode data.
- Prefix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065. A value of 000 means the end of the prefix. The reader will not process any prefix charcters after a 000. A null, therefore, cannot be entered in as a prefix character. The prefix 065 066 067 000 000 is valid, and will prepend the barcode with "ABC". The prefix sequence 065 066 000 067 000 will prepend only an "AB" because the 000 signifies the end.
- The largest decimal value (ascii char) that can be used as a prefix is 126.

Industrial 2 of 5 Barcode Suffix

X031 aaa bbb ccc ddd eee

aaa= suffix character 1	3 digit ascii character, 000 denotes end
bbb= suffix character 2	3 digit ascii character, 000 denotes end
ccc= suffix character 3	3 digit ascii character, 000 denotes end
ddd= suffix character 4	3 digit ascii character, 000 denotes end
eee= suffix character 5	3 digit ascii character, 000 denotes end

notes

- Up to 5 suffix characters can be programmed for ind2of5 barcodes. If the suffix option is turned on, then these suffix characters are appended to the ind2of5 barcode data.
- Suffix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065. a value of 000 means the end of the suffix. The reader will not process any suffix charcters after a 000. A null, therefore, cannot be entered in as a suffix character. The suffix 065 066 067 000 000 is valid, and will place "ABC" after the barcode. The suffix sequence 065 066 000 067 000 will place only an "AB" after the barcode because the 000 signifies the end.

- The largest decimal value (ascii char) that can be used as a suffix is 126.

Interleaved 2 of 5 General Setup

X004 e c t d ll x p s

e	=	enable I2of5	0=disabled, 1=enabled
c	=	check digit	0=disabled, 1=enabled
t	=	transmit check digit	0=disabled, 1=enabled
d	=	transmit direction indicator	0=disabled, 1=enabled
ll	=	length	restrict length of barcode to ll
x	=	extra bars	# extra bars allowed
p	=	prefix	0=disabled, 1=enabled
s	=	suffix	0=disabled, 1=enabled

notes

- e parameter must be 1 in order to read I2of5 barcodes
- If p is set to a 1, then the programmed prefix is prepended to the barcode data. Note that the prefix can be up to 5 characters in length, and is prepended to the I2of5 barcode data after masking has taken place, if any.
- If c is set to a 1, then the I2of5 barcode you are reading must have a check digit in it in order to be valid. The check digit is not transmitted unless the t parameter is set to a 1.
- If d is set to a 1, then a direction indicator is prepended to barcode data. The direction indicator is either an "f" for forward scan, or "r" for reverse scan.
- If s (suffix) is set to a 1, a suffix of up to 5 characters may be added to the end of the barcode.
- The ll parameter specifies the length of the barcode to read. If length is set to a non zero value, then the I2of5 barcode read must match that length or it will be ignored.
- The x parameter refers to the number of allowable extra bars that the reader can use when attempting to decode a I2of5 barcode. It is suggested that this parameter be left at the default of 2.

Interleaved 2 of 5 Barcode Prefix

X018 aaa bbb ccc ddd eee

aaa= prefix character 1	3 digit ascii character, 000 denotes end
bbb= prefix character 2	3 digit ascii character, 000 denotes end
ccc= prefix character 3	3 digit ascii character, 000 denotes end
ddd= prefix character 4	3 digit ascii character, 000 denotes end
eee= prefix character 5	3 digit ascii character, 000 denotes end

notes

- Up to 5 prefix characters can be programmed for I2of5 barcodes. If the prefix option is turned on, then these prefix characters are prepended to the I2of5 barcode data.
- Prefix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065. A value of 000 means the end of the prefix. The reader will not process any prefix charcters after a 000. A null, therefore, cannot be entered in as a prefix character. The prefix 065 066 067 000 000 is valid, and will prepend the barcode with "ABC". The prefix sequence 065 066 000 067 000 will prepend only an "AB" because the 000 signifies the end.
- The largest decimal value (ascii char) that can be used as a prefix is 126.

Interleaved 2 of 5 Barcode Suffix

X019 aaa bbb ccc ddd eee

aaa= suffix character 1	3 digit ascii character, 000 denotes end
bbb= suffix character 2	3 digit ascii character, 000 denotes end
ccc= suffix character 3	3 digit ascii character, 000 denotes end
ddd= suffix character 4	3 digit ascii character, 000 denotes end
eee= suffix character 5	3 digit ascii character, 000 denotes end

notes

- Up to 5 suffix characters can be programmed for I2of5 barcodes. If the suffix option is turned on, then these suffix characters are appended to the I2of5 barcode data.
- Suffix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065. A value of 000 means the end of the suffix. The reader will not process any suffix charcters after a 000. A null, therefore, cannot be entered in as a suffix character. The suffix 065 066 067 000 000 is valid, and will place "ABC" after the barcode. The suffix sequence 065 066 000 067 000 will place only an "AB" after the barcode because the 000 signifies the end.
- The largest decimal value (ascii char) that can be used as a suffix is 126.

UPC-A General Setup

X005 e d x p s

e	= enable I2of5	0=disabled, 1=enabled
d	= transmit direction indicator	0=disabled, 1=enabled
x	= extra bars	# extra bars allowed
p	= prefix	0=disabled, 1=enabled
s	= suffix	0=disabled, 1=enabled

notes

- e parameter must be 1 in order to read UPC-A barcodes
- If p is set to a 1, then the programmed prefix is prepended to the barcode data. Note that the prefix can be up to 5 characters in length, and is prepended to the UPC-A barcode data after masking has taken place, if any.
- If d is set to a 1, then a direction indicator is prepended to barcode data. The direction indicator is either an "f" for forward scan, or "r" for reverse scan.
- If s (suffix) is set to a 1, a suffix of up to 5 characters may be added to the end of the barcode.
- The x parameter refers to the number of allowable extra bars that the reader can use when attempting to decode a UPC-A barcode. It is suggested that this parameter be left at the default of 2.

UPC-A Barcode Prefix

X026 aaa bbb ccc ddd eee

aaa=	prefix character 1	3 digit ascii character, 000 denotes end
bbb=	prefix character 2	3 digit ascii character, 000 denotes end
ccc=	prefix character 3	3 digit ascii character, 000 denotes end
ddd=	prefix character 4	3 digit ascii character, 000 denotes end
eee=	prefix character 5	3 digit ascii character, 000 denotes end

notes

- Up to 5 prefix characters can be programmed for UPC-A barcodes. If the prefix option is turned on, then these prefix characters are prepended to the UPC-A barcode data.
- Prefix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065. A value of 000 means the end of the prefix. The reader will not process any prefix charcters after a 000. A null, therefore, cannot be entered in as a prefix character. The prefix 065 066 067 000 000 is valid, and will prepend the barcode with "ABC". The prefix sequence 065 066 000 067 000 will prepend only an "AB" because the 000 signifies the end.
- The largest decimal value (ascii char) that can be used as a prefix is 126.

UPC-A Barcode Suffix

X027 aaa bbb ccc ddd eee

aaa= suffix character 1	3 digit ascii character, 000 denotes end
bbb= suffix character 2	3 digit ascii character, 000 denotes end
ccc= suffix character 3	3 digit ascii character, 000 denotes end
ddd= suffix character 4	3 digit ascii character, 000 denotes end
eee= suffix character 5	3 digit ascii character, 000 denotes end

notes

- Up to 5 suffix characters can be programmed for UPC-A barcodes. If the suffix option is turned on, then these suffix characters are appended to the UPC-A barcode data.
- Suffix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065. a value of 000 means the end of the suffix. The reader will not process any suffix charcters after a 000. A null, therefore, cannot be entered in as a suffix character. The suffix 065 066 067 000 000 is valid, and will place "ABC" after the barcode. The suffix sequence 065 066 000 067 000 will place only an "AB" after the barcode because the 000 signifies the end.
- The largest decimal value (ascii char) that can be used as a suffix is 126.

EAN-13 General Setup

X006 e d x p s

e = enable EAN13	0=disabled, 1=enabled
d = transmit direction indicator	0=disabled, 1=enabled
x = extra bars	# extra bars allowed
p = prefix	0=disabled, 1=enabled
s = suffix	0=disabled, 1=enabled

notes

- e parameter must be 1 in order to read EAN13 barcodes
- If p is set to a 1, then the programmed prefix is prepended to the barcode data. Note that the prefix can be up to 5 characters in length, and is prepended to the EAN13 barcode data after masking has taken place, if any.
- If d is set to a 1, then a direction indicator is prepended to barcode data. The direction indicator is either an "f" for forward scan, or "r" for reverse scan.
- If s (suffix) is set to a 1, a suffix of up to 5 characters may be added to the end of the barcode.
- The x parameter refers to the number of allowable extra bars that the reader can use when attempting to decode a EAN-13 barcode. It is suggested that this parameter be left at the default of 2.

EAN13 Barcode Prefix

X028 aaa bbb ccc ddd eee

aaa= prefix character 1	3 digit ascii character, 000 denotes end
bbb= prefix character 2	3 digit ascii character, 000 denotes end
ccc= prefix character 3	3 digit ascii character, 000 denotes end
ddd= prefix character 4	3 digit ascii character, 000 denotes end
eee= prefix character 5	3 digit ascii character, 000 denotes end

notes

- Up to 5 prefix characters can be programmed for EAN13 barcodes. If the prefix option is turned on, then these prefix characters are prepended to the EAN13 barcode data.
- Prefix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065. A value of 000 means the end of the prefix. The reader will not process any prefix charcters after a 000. A null, therefore, cannot be entered in as a prefix character. The prefix 065 066 067 000 000 is valid, and will prepend the barcode with "ABC". The prefix sequence 065 066 000 067 000 will prepend only an "AB" because the 000 signifies the end.
- The largest decimal value (ascii char) that can be used as a prefix is 126.

EAN13 Barcode Suffix

X029 aaa bbb ccc ddd eee

aaa= suffix character 1	3 digit ascii character, 000 denotes end
bbb= suffix character 2	3 digit ascii character, 000 denotes end
ccc= suffix character 3	3 digit ascii character, 000 denotes end
ddd= suffix character 4	3 digit ascii character, 000 denotes end
eee= suffix character 5	3 digit ascii character, 000 denotes end

notes

- Up to 5 suffix characters can be programmed for EAN13 barcodes. If the suffix option is turned on, then these suffix characters are appended to the EAN13 barcode data.
- Suffix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065. A value

of 000 means the end of the suffix. The reader will not process any suffix charcters after a 000. A null, therefore, cannot be entered in as a suffix character. The suffix 065 066 067 000 000 is valid, and will place "ABC" after the barcode. The suffix sequence 065 066 000 067 000 will place only an "AB" after the barcode because the 000 signifies the end.

- The largest decimal value (ascii char) that can be used as a suffix is 126.

Code128 General Setup

X007 e c t d ll x p s

e	=	enable Code128	0=disabled, 1=enabled
c	=	check digit	0=disabled, 1=enabled
t	=	transmit check digit	0=disabled, 1=enabled
d	=	transmit direction indicator	0=disabled, 1=enabled
ll	=	restrict length	00=any, ll=restrict length
x	=	extra bars	# extra bars allowed
p	=	prefix	0=disabled, 1=enabled
s	=	suffix	0=disabled, 1=enabled

notes

- e parameter must be 1 in order to read Code128 barcodes
- c parameter is set to a 1 if check digit validtion is required
- t parmeter is set to a 1 if check digit is to be transmitted
- Set ll to 00 to allow any length (default) or to a 2 digit length to restrict the length
- If p is set to a 1, then the programmed prefix is prepended to the barcode data. Note that the prefix can be up to 5 characters in length, and is prepended to the barcode data after masking has taken place, if any.
- If d is set to a 1, then a direction indicator is prepended to barcode data. The direction indicator is either an "f" for forward scan, or "r" for reverse scan.
- If s (suffix) is set to a 1, a suffix of up to 5 characters may be added to the end of the barcode.
- The x parameter refers to the number of allowable extra bars that the reader can use when attempting to decode a Code 128 barcode. It is suggested that this parameter be left at the default of 2.

Code128 Barcode Prefix

X024 aaa bbb ccc ddd eee

aaa= prefix character 1	3 digit ascii character, 000 denotes end
bbb= prefix character 2	3 digit ascii character, 000 denotes end
ccc= prefix character 3	3 digit ascii character, 000 denotes end
ddd= prefix character 4	3 digit ascii character, 000 denotes end
eee= prefix character 5	3 digit ascii character, 000 denotes end

notes

- Up to 5 prefix characters can be programmed for Code128 barcodes. If the prefix option is turned on, then these prefix characters are prepended to the barcode data.
- Prefix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065. A value of 000 means the end of the prefix. The reader will not process any prefix charcters after a 000. A null, therefore, cannot be entered in as a prefix character. The prefix 065 066 067 000 000 is valid, and will prepend the barcode with "ABC". The prefix sequence 065 066 000 067 000 will prepend only an "AB" because the 000 signifies the end.
- The largest decimal value (ascii char) that can be used as a prefix is 126.

Code128 Barcode Suffix

X025 aaa bbb ccc ddd eee

aaa= suffix character 1	3 digit ascii character, 000 denotes end
bbb= suffix character 2	3 digit ascii character, 000 denotes end
ccc= suffix character 3	3 digit ascii character, 000 denotes end
ddd= suffix character 4	3 digit ascii character, 000 denotes end
eee= suffix character 5	3 digit ascii character, 000 denotes end

notes

- Up to 5 suffix characters can be programmed for Code128 barcodes. If the suffix option is turned on, then these suffix characters are appended to the barcode data.
- Suffix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065. A value of 000 means the end of the suffix. The reader will not process any suffix charcters after a 000. A null, therefore, cannot be entered in as a suffix character. The suffix 065 066 067 000 000 is valid, and will place "ABC" after the barcode. The suffix sequence 065 066 000 067 000 will place only an "AB" after the barcode because the 000 signifies the end.
- The largest decimal value (ascii char) that can be used as a suffix is 126.

Codabar General Setup

X008 e c t s d ll x i p s

e	=	enable Codabar	0=disabled, 1=enabled
c	=	check digit	0=disabled, 1=enabled
t	=	transmit check digit	0=disabled, 1=enabled
s	=	transmit start/stop	0=disabled, 1=enabled
d	=	transmit direction indicator	0=disabled, 1=enabled
ll	=	restrict length	00=any, ll=restrict length
x	=	extra bars	# extra bars allowed
i	=	clsi	0=disabled, 1=enabled
p	=	prefix	0=disabled, 1=enabled
s	=	suffix	0=disabled, 1=enabled

notes

- e parameter must be 1 in order to read Codabar barcodes
- c parameter is set to a 1 if check digit validtion is required
- t parmeter is set to a 1 if check digit is to be transmitted
- Set s to a 1 to transmit the start and stop characters
- Set i to a 1 to enable clsi interpretation
- Set ll to 00 to allow any length (default) or to a 2 digit length to restrict the length
- If p is set to a 1, then the programmed prefix is prepended to the barcode data. Note that the prefix can be up to 5 characters in length, and is prepended to the barcode data after masking has taken place, if any.
- If d is set to a 1, then a direction indicator is prepended to barcode data. The direction indicator is either an "f" for forward scan, or "r" for reverse scan.
- If s (suffix) is set to a 1, a suffix of up to 5 characters may be added to the end of the barcode.
- The x parameter refers to the number of allowable extra bars that the reader can use when attempting to decode a Codabar barcode. It is suggested that this parameter be left at the default of 2.

Codabar Barcode Prefix

X022 aaa bbb ccc ddd eee

aaa=	prefix character 1	3 digit ascii character, 000 denotes end
bbb=	prefix character 2	3 digit ascii character, 000 denotes end
ccc=	prefix character 3	3 digit ascii character, 000 denotes end
ddd=	prefix character 4	3 digit ascii character, 000 denotes end
eee=	prefix character 5	3 digit ascii character, 000 denotes end

notes

- Up to 5 prefix characters can be programmed for Codabar barcodes. If the prefix option is turned on, then these prefix characters are prepended to the barcode data.
- Prefix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065. A value of 000 means the end of the prefix. The reader will not process any prefix charcters after a 000. A null, therefore, cannot be entered in as a prefix character. The prefix 065 066 067 000 000 is valid, and will prepend the barcode with "ABC". The prefix sequence 065 066 000 067 000 will prepend only an "AB" because the 000 signifies the end.
- The largest decimal value (ascii char) that can be used as a prefix is 126.

Codabar Barcode Suffix

X023 aaa bbb ccc ddd eee

aaa= suffix character 1	3 digit ascii character, 000 denotes end
bbb= suffix character 2	3 digit ascii character, 000 denotes end
ccc= suffix character 3	3 digit ascii character, 000 denotes end
ddd= suffix character 4	3 digit ascii character, 000 denotes end
eee= suffix character 5	3 digit ascii character, 000 denotes end

notes

- Up to 5 suffix characters can be programmed for Codabar barcodes. If the suffix option is turned on, then these suffix characters are appended to the barcode data.
- Suffix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065. A value of 000 means the end of the suffix. The reader will not process any suffix charcters after a 000. A null, therefore, cannot be entered in as a suffix character. The suffix 065 066 067 000 000 is valid, and will place "ABC" after the barcode. The suffix sequence 065 066 000 067 000 will place only an "AB" after the barcode because the 000 signifies the end.
- The largest decimal value (ascii char) that can be used as a suffix is 126.

Barcode Mask #1 Command - Part 1

X033 ll p1 ll1 p2 ll2

ll = length	length to match, 00 = any length
p1 = posistion 1	start position, 00 means ll1 is a constant
ll1 = length/constant 1	length,000=take remaining, or constant value
p2 = posistion 2	
ll2 = length/constant 2	

notes

- There are 2 possible masks for barcodes, which are executed if the length of the barcode matches the length specified in the ll parameter.
- There are two commands needed to program a mask. They are broken up into two parts to make the command shorter. Both commands need to be entered in to properly program a mask.
- The barcode mask can specify a substring to extract. In this case, p1 and p2 specify the starting positions of the substrings, and ll1,ll2 specify the number of characters to extract. If ll1,ll2 are 000 then the remainder of the barcode (starting at the p1,p2 position) is extracted.
- If p1,p2 is 00, the ll1 or ll2 refer to an ascii chararacter which is inserted at the current position. Ascii characters are entered in using their 3 digit decimal value.

Barcode Mask #1 Command - Part 2

X034 p3 ll3 p4 ll4 p5 ll5

p3 = posistion 1 — parameters same as above
ll3 = length/constant 1
p4 = posistion 2
ll4 = length/constant 2
p5 = posistion 2
ll5 = length/constant 2

notes

- There are 2 possible masks for barcodes, which are executed if the length of the barcode matches the length specified in the ll parameter.
- There are two command needed to program a mask. They are broken up into two parts to make the command shorter. Both commands need to be entered in to properly program a mask.
- The barcode mask can specify a substring to extract. In this case, p3,p4, and p5 specify the starting positions of the substrings, and ll3,ll4, and ll5 specify the number of characters to extract. If ll3,ll4,ll5 are 000 then the remainder of the barcode (starting at the p3,p4, or p5 position) is extracted.
- If p3,p4, or p5 is 00, then ll3,ll4,ll5 refer to an ascii chararacter which is inserted at the current position. Ascii characters are entered in using their 3 digit decimal value.

Barcode Mask #2 Command - Part 1

X035 ll p1 ll1 p2 ll2

ll = length — length to match, 00 = any length
p1 = posistion 1 — start position, 00 means ll1 is a constant
ll1 = length/constant 1 — length,000=take remaining, or constant value
p2 = posistion 2
ll2 = length/constant 2

notes

- There are 2 possible masks for barcodes, which are executed if the length of the barcode matches the length specified in the ll parameter.
- There are two commands needed to program a mask. They are broken up into two parts to make the command shorter. Both commands need to be entered in to properly program a mask.
- The barcode mask can specify a substring to extract. In this case, p1 and p2 specify the starting positions of the substrings, and ll1,ll2 specify the number of characters to extract. If ll1,ll2 are 000 then the remainder of the barcode (starting at the p1,p2 position) is extracted.
- If p1,p2 is 00, the ll1 or ll2 refer to an ascii chararacter which is inserted at the current position. Ascii characters are entered in using their 3 digit decimal value.

Barcode Mask #2 Command - Part 2

X036 p3 ll3 p4 ll4 p5 ll5

p3 = posistion 1
ll3 = length/constant 1
p4 = posistion 2
ll4 = length/constant 2
p5 = posistion 2
ll5 = length/constant 2

parameters same as above

notes

- There are 2 possible masks for barcodes, which are executed if the length of the barcode matches the length specified in the ll parameter.
- There are two command needed to program a mask. They are broken up into two parts to make the command shorter. Both commands need to be entered in to properly program a mask.
- The barcode mask can specify a substring to extract. In this case, p3,p4, and p5 specify the starting positions of the substrings, and ll3,ll4, and ll5 specify the number of characters to extract. If ll3,ll4,ll5 are 000 then the remainder of the barcode (starting at the p3,p4, or p5 position) is extracted.
- If p3,p4, or p5 is 00, then ll3,ll4,ll5 refer to an ascii chararacter which is inserted at the current position. Ascii characters are entered in using their 3 digit decimal value.

USID/CAC Barcode Programming

Stax readers can read and decode 24 character and 21 character Usid cards, 18 character Usid cards, and 18 character Cac cards. Note that Usid/Cac must be explicitly turned on in the barcode general options.

Usid21 Processing Options

X104 1 2 3 4 5 6 7 8

1= Process active type A	0=no, 1=yes
2= Process retired type B	0=no, 1=yes
3= Process reserve Type C	0=no, 1=yes
4= Process sponsor Type D	0=no, 1=yes
5= Process dependent type E,F,G,H	0=no, 1=yes
6= Process civilian type K,L,M	0=no, 1=yes
7= Process retired reserve type I,J	0=no, 1=yes
8= Data type for dependents	0=dependent pdi, 1=sponsor pdi

notes

- To process a particular card type, the type must be explicitly turned on.
- The data type used for dependent cards can be the dependent's pdi, or the sponsor's pdi.

Usid18/Cac Processing Options (firmware prior to May 20, 2011)

X105 1 2 3 4

1= Process all except dependents	0=no, 1=yes
2= Process dependents	0=no, 1=yes
3= datatype to process (non dependent)	0=pdi, 1=edi
4= datatype to process (dependent)	0=dependent edi, 1=sponsor's edi

notes

- To process a particular card type, the type must be explicitly turned on.
- The data type used for dependent cards can be the dependent's pdi, or the sponsor's pdi.

Usid18/Cac Processing Options (firmware after May 20, 2011)

X105 1 2 3 4 5 6

1= Process all except dependents	0=no, 1=yes
2= Process dependents	0=no, 1=yes
3= datatype to process (non dependent)	0=pdi, 1=edi
4= datatype to process (dependent)	0=dependent edi, 1=sponsor's edi
5= append card instance identifier	0=no, 1=yes
6= transpose card instance identifier	0=no, 1=yes

notes

- To process a particular card type, the type must be explicitly turned on.
- The data type used for dependent cards can be the dependent's edi, or the sponsor's edi.
- If parameter 5 is yes (append card instance), then the card instance (CI) is read and processed. There are two ways to process the CI. The first method decodes the CI as a 3-digit number, representing the decimal value of the ascii CI. For example, a CI of "A" is converted to a decimal 065. The second method assigns a decimal value to the CI based on it's position of the 36 possible CI values, i.e "0" = 0, "A" = 10, "Z" = 35. This value is decoded into a 2 digit number. To use the second method, set parameter 6 to "yes". The CI value will then be appended to the EDI or PDI number. For non-serial readers (wiegand, magstripe) the appropriate fields must be set up to extract the proper lengths for the PDI or EDI. The remaining characters will comprise of the CI, which should be put into the Issue Code field for subsequent transmission.
- For wiegand outputs, the extracted ID data is placed into the ID field.
- For wiegand output transmissions, we suggest using 34 bits, no parity, for the ID field. Transmit only the ID field. If the CI append is on, then 42 bits will be transmitted in total. Usid18/Cac Processing Options (firmware after May 20, 2011)

PDI Output Options (firmware after June 26, 2011)

X120 1 2 3

1= PDI Length	0=9, 1=10
2= Append 000 for an Issue Code	0=no, 1=yes
3= Append 00 for an Issue Code	0=no, 1=yes

notes

- The output PDI length can be either 9 or 10. Only 9 digits are needed, however 10 can be selected to provide compatability in systems that are also reading EDI's.
- Either 00 or 000 can be appended to the PDI data, to provide compatability with 18 character cards that process the CI. In this case, the 00 will be treated as the Issue Code.

Magstripe Programming Commands

Magstripe programming consists of a number of commands. There is one General Setup command specifying global options for magstripe processing. Each Track also has a general options command specific to that track, as well as commands to program prefixes and suffixes for each track.

General Magstripe Options Command

X045 1 2

1	=	tk1 required	0=no, 1=yes
2	=	tk2 required	0=no, 1=yes

notes

- The "required" parameters are set to a 1 if you want to ignore all reads which do not have a valid decode on those tracks which you specified were required.
- **** Please Note - Track3 processing is not available at this time.

Magstripe Track 1 settings - Part 1

X039 e c y s l d a m b p x

e	=	track 1 enabled	0=no, 1=yes
c	=	use track 1 character set	0=no, 1=yes
y	=	c start	0=no, 1=yes
s	=	include start/stop	0=no, 1=yes
l	=	include lrc	0=no, 1=yes
d	=	direction indicator	0=no, 1=yes
a	=	alpha delete	0=no, 1=yes
m	=	masking	0=off, 1=on
b	=	good read beep	0=off, 1=on
p	=	prefix	0=off, 1=on
s	=	suffix	0=off, 1=on

notes

- Setting "e" to a 1 enables the processing of track 1 data. The "enable" is not the same as the "required" setting

in the general magstripe options command. The required option means that the track is required and must be properly decoded and valid. The enabled option means to process the track data after decoding. For example, you may wish to read and process data read on track 2 of a magstripe card, but only want to process the track 2 data if there is also valid data on track 1. To do this, you set both tracks1 and 2 as required, but only enable track 2.

- The "c" option specified whether the track 1 character set is to be used when decoding the magstripe track.
- Setting "y" to a 1 tells the reader to only process the data if the data starts with a c start rather than the traditional b start. This applies only to the track 2 character set.
- If "s" is a 1, then the start and stop characters are included with the decoded data.
- If "l" is set to a 1 then the lrc character is included with the decoded data.
- If "d" is a 1, then a direction indicator (f for forward, r for reverse scan) is prepended to the decoded data.
- If "a" is a 1 then all alphas are deleted from the decoded data.
- If "M" is a 1 then the magstripe data is masked after it is read.
- If "b" is set to a 1 then the reader makes a beep with a proper decode.
- If "p" is set to a 1 then up to 5 programmable prefix characters can be prepended to the decoded data.
- If "s" is set to a 1 then up to 5 programmable suffix characters can be appended to the decoded data.

Magstripe Track 1 settings - Part 2

X040 ll ss

ll	=	track 1 length	ll=length, 00=any length
ss	=	seperator char translate	3 digit ascii value, 000=no translate

notes

- If ll is set to 00, then any length track 1 magstripe may be read. If ll is set to a specific length, then only track1 dat with that length will be read.
- Set sss to the 3-digit decimal value of any ascii character that you want to use to replace any seperator characters in the magstripe data.

Magstripe Track 2 settings - Part 1

X041 e c y s l d a m b p x

e	=	track 2 enabled	0=no, 1=yes
c	=	use track 1 character set	0=no, 1=yes
y	=	c start	0=no, 1=yes
s	=	include start/stop	0=no, 1=yes
l	=	include lrc	0=no, 1=yes
d	=	direction indicator	0=no, 1=yes
a	=	alpha delete	0=no, 1=yes
m	=	masking	0=off, 1=on
b	=	good read beep	0=off, 1=on
p	=	prefix	0=off, 1=on
s	=	suffix	0=off, 1=on

notes

- Setting "e" to a 1 enables the processing of track 2 data. The "enable" is not the same as the "required" setting in the general magstripe options command. The required option means that the track is required and must be properly decoded and valid. The enabled option means to process the track data after decoding. For example, you may wish to read and process data read on track 2 of a magstripe card, but only want to process the track 2 data if there is also valid data on track 1. To do this, you set both tracks1 and 2 as required, but only enable track 2.
- The "c" option specified whether the track 1 character set is to be used when decoding the magstripe track.
- Setting "y" to a 1 tells the reader to only process the data if the data starts with a c start rather than the traditional b start. This applies only to the track 2 character set.
- If "s" is a 1, then the start and stop characters are included with the decoded data.
- If "l" is set to a 1 then the lrc character is included with the decoded data.
- If "d" is a 1, then a direction indicator (f for forward, r for reverse scan) is prepended to the decoded data.
- If "a" is a 1 then all alphas are deleted from the decoded data.
- If "m" is a 1 then the magstripe data is masked after it is read.
- If "b" is set to a 1 then the reader makes a beep with a proper decode.
- If "p" is set to a 1 then up to 5 programmable prefix characters can be prepended to the decoded data.
- If "s" is set to a 1 then up to 5 programmable suffix characters can be appended to the decoded data.

Magstripe Track 2 settings - Part 2

X042 ll ss

ll	= track 2 length	ll=length, 00=any length
ss	= seperator char translate	3 digit ascii value, 000=no translate

notes

- If ll is set to 00, then any length track 2 magstripe may be read. If ll is set to a specific length, then only track2 dat with that length will be read.
- Set sss to the 3-digit decimal value of any ascii character that you want to use to replace any seperator characters in the magstripe data.

Track 1 Prefix

X055 aaa bbb ccc ddd eee

aaa=	prefix character 1	3 digit ascii character, 000 denotes end
bbb=	prefix character 2	3 digit ascii character, 000 denotes end
ccc=	prefix character 3	3 digit ascii character, 000 denotes end
ddd=	prefix character 4	3 digit ascii character, 000 denotes end
eee=	prefix character 5	3 digit ascii character, 000 denotes end

notes

- Up to 5 prefix characters can be programmed for Track 1 data. If the prefix option is turned on, then these prefix characters are prepended to the track 1 data.
- Prefix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065. a value of 000 means the end of the prefix. The reader will not process any prefix charcters after a 000. A null, therefore, cannot be entered in as a prefix character. The prefix 065 066 067 000 000 is valid, and will prepend the track 1 data with "ABC". The prefix sequence 065 066 000 067 000 will prepend only an "AB" because the 000 signifies the end.
- The largest decimal value (ascii char) that can be used as a prefix is 126.

Track 1 Suffix

X058 aaa bbb ccc ddd eee

aaa= suffix character 1	3 digit ascii character, 000 denotes end
bbb= suffix character 2	3 digit ascii character, 000 denotes end
ccc= suffix character 3	3 digit ascii character, 000 denotes end
ddd= suffix character 4	3 digit ascii character, 000 denotes end
eee= suffix character 5	3 digit ascii character, 000 denotes end

notes

- Up to 5 suffix characters can be programmed for Track 1 If the suffix option is turned on, then these suffix characters are appended to the track 1 data.
- Suffix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065. A value of 000 means the end of the suffix. The reader will not process any suffix charcters after a 000. A null, therefore, cannot be entered in as a suffix character. The suffix 065 066 067 000 000 is valid, and will place "ABC" after the barcode. The suffix sequence 065 066 000 067 000 will place only an "AB" after the magstripe because the 000 signifies the end.
- The largest decimal value (ascii char) that can be used as a suffix is 126.

Track 2 Prefix

X056 aaa bbb ccc ddd eee

aaa= prefix character 1	3 digit ascii character, 000 denotes end
bbb= prefix character 2	3 digit ascii character, 000 denotes end
ccc= prefix character 3	3 digit ascii character, 000 denotes end
ddd= prefix character 4	3 digit ascii character, 000 denotes end
eee= prefix character 5	3 digit ascii character, 000 denotes end

notes

- Up to 5 prefix characters can be programmed for Track 2 data. If the prefix option is turned on, then these prefix characters are prepended to the track 2 data.
- Prefix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065. a value of 000 means the end of the prefix. The reader will not process any prefix charcters after a 000. A null, therefore, cannot be entered in as a prefix character. The prefix 065 066 067 000 000 is valid, and will prepend the track 2 data with "ABC". The prefix sequence 065 066 000 067 000 will prepend only an "AB" because the 000 signifies the end.
- The largest decimal value (ascii char) that can be used as a prefix is 126.

Track 2 Suffix

X059 aaa bbb ccc ddd eee

aaa= suffix character 1	3 digit ascii character, 000 denotes end
bbb= suffix character 2	3 digit ascii character, 000 denotes end
ccc= suffix character 3	3 digit ascii character, 000 denotes end
ddd= suffix character 4	3 digit ascii character, 000 denotes end
eee= suffix character 5	3 digit ascii character, 000 denotes end

notes

- Up to 5 suffix characters can be programmed for Track 2 If the suffix option is turned on, then these suffix characters are appended to the track 2 data.
- Suffix characters are entered in as a 3 digit ascii number, representing the decimal number of the ascii code. For example, "A" is a hex 41, decimal 065. A value of 000 means the end of the suffix. The reader will not process any suffix charcters after a 000. A null, therefore, cannot be entered in as a suffix character. The suffix 065 066 067 000 000 is valid, and will place "ABC" after the barcode. The suffix sequence 065 066 000 067 000 will place only an "AB" after the magstripe because the 000 signifies the end.
- The largest decimal value (ascii char) that can be used as a suffix is 126.

Track 1 Mask #1 Command - Part 1

X061 ll p1 ll1 p2 ll2

ll = length	length to match, 00 = any length
p1 = posistion 1	start position, 00 means ll1 is a constant
ll1 = length/constant 1	length,000=take remaining, or constant value
p2 = posistion 2	
ll2 = length/constant 2	

notes

- There are 2 possible masks for track 1 data, which are executed if the length of the track 1 data matches the length specified in the ll parameter.
- There are two commands needed to program a mask. They are broken up into two parts to make the command shorter. Both commands need to be entered in to properly program a mask.
- The track 1 mask can specify a substring to extract. In this case, p1 and p2 specify the starting positions of the substrings, and ll1,ll2 specify the number of characters to extract. If ll1,ll2 are 000 then the remainder of the track 1 data (starting at the p1,p2 position) is extracted.
- If p1,p2 is 00, the ll1 or ll2 refer to an ascii chararacter which is inserted at the current position. Ascii characters are entered in using their 3 digit decimal value.

Track 1 Mask #1 Command - Part 2

X062 p3 ll3 p4 ll4 p5 ll5

p3 = posistion 1
ll3 = length/constant 1
p4 = posistion 2
ll4 = length/constant 2
p5 = posistion 2
ll5 = length/constant 2

parameters same as above

notes

- There are 2 possible masks for track 1, which are executed if the length of the track 1 data matches the length specified in the ll parameter.
- There are two command needed to program a mask. They are broken up into two parts to make the command shorter. Both commands need to be entered in to properly program a mask.
- The track 1 mask can specify a substring to extract. In this case, p3,p4, and p5 specify the starting positions of the substrings, and ll3,ll4, and ll5 specify the number of characters to extract. If ll3,ll4,ll5 are 000 then the remainder of the track 1 data (starting at the p3,p4, or p5 position) is extracted.
- If p3,p4, or p5 is 00, then ll3,ll4,ll5 refer to an ascii chararacter which is inserted at the current position. Ascii characters are entered in using their 3 digit decimal value.

Track 1 Mask #2 Command - Part 1

X063 ll p1 ll1 p2 ll2

ll = length — length to match, 00 = any length
p1 = posistion 1 — start position, 00 means ll1 is a constant
ll1 = length/constant 1 — length,000=take remaining, or constant value
p2 = posistion 2
ll2 = length/constant 2

notes

- There are 2 possible masks for track 1 data, which are executed if the length of the track 1 data matches the length specified in the ll parameter.
- There are two commands needed to program a mask. They are broken up into two parts to make the command shorter. Both commands need to be entered in to properly program a mask.
- The track 1 mask can specify a substring to extract. In this case, p1 and p2 specify the starting positions of the substrings, and ll1,ll2 specify the number of characters to extract. If ll1,ll2 are 000 then the remainder of the track 1 data (starting at the p1,p2 position) is extracted.
- If p1,p2 is 00, the ll1 or ll2 refer to an ascii chararacter which is inserted at the current position. Ascii characters are entered in using their 3 digit decimal value.

Track 1 Mask #2 Command - Part 2

X064 p3 ll3 p4 ll4 p5 ll5

p3 = posistion 1
ll3 = length/constant 1
p4 = posistion 2
ll4 = length/constant 2
p5 = posistion 2
ll5 = length/constant 2

parameters same as above

notes

- There are 2 possible masks for track 1, which are executed if the length of the track 1 data matches the length specified in the ll parameter.
- There are two command needed to program a mask. They are broken up into two parts to make the command shorter. Both commands need to be entered in to properly program a mask.
- The track 1 mask can specify a substring to extract. In this case, p3,p4, and p5 specify the starting positions of the substrings, and ll3,ll4, and ll5 specify the number of characters to extract. If ll3,ll4,ll5 are 000 then the remainder of the track 1 data (starting at the p3,p4, or p5 position) is extracted.
- If p3,p4, or p5 is 00, then ll3,ll4,ll5 refer to an ascii chararacter which is inserted at the current position. Ascii characters are entered in using their 3 digit decimal value.

Track 2 Mask #1 Command - Part 1

X065 ll p1 ll1 p2 ll2

ll = length — length to match, 00 = any length
p1 = posistion 1 — start position, 00 means ll1 is a constant
ll1 = length/constant 1 — length,000=take remaining, or constant value
p2 = posistion 2
ll2 = length/constant 2

notes

- There are 2 possible masks for track 2 data, which are executed if the length of the track 2 data matches the length specified in the ll parameter.
- There are two commands needed to program a mask. They are broken up into two parts to make the command shorter. Both commands need to be entered in to properly program a mask.
- The track 2 mask can specify a substring to extract. In this case, p1 and p2 specify the starting positions of the substrings, and ll1,ll2 specify the number of characters to extract. If ll1,ll2 are 000 then the remainder of the track 2 data (starting at the p1,p2 position) is extracted.
- If p1,p2 is 00, the ll1 or ll2 refer to an ascii chararacter which is inserted at the current position. Ascii characters are entered in using their 3 digit decimal value.

Track 2 Mask #1 Command - Part 2

X066 p3 ll3 p4 ll4 p5 ll5

p3	= posistion 1	parameters same as above
ll3	= length/constant 1	
p4	= posistion 2	
ll4	= length/constant 2	
p5	= posistion 2	
ll5	= length/constant 2	

notes

- There are 2 possible masks for track 2, which are executed if the length of the track 2 data matches the length specified in the ll parameter.
- There are two command needed to program a mask. They are broken up into two parts to make the command shorter. Both commands need to be entered in to properly program a mask.
- The track 2 mask can specify a substring to extract. In this case, p3,p4, and p5 specify the starting positions of the substrings, and ll3,ll4, and ll5 specify the number of characters to extract. If ll3,ll4,ll5 are 000 then the remainder of the track 2 data (starting at the p3,p4, or p5 position) is extracted.
- If p3,p4, or p5 is 00, then ll3,ll4,ll5 refer to an ascii chararacter which is inserted at the current position. Ascii characters are entered in using their 3 digit decimal value.

Track 2 Mask #2 Command - Part 1

X067 ll p1 ll1 p2 ll2

ll	= length	length to match, 00 = any length
p1	= posistion 1	start position, 00 means ll1 is a constant
ll1	= length/constant 1	length,000=take remaining, or constant value
p2	= posistion 2	
ll2	= length/constant 2	

notes

- There are 2 possible masks for track 2 data, which are executed if the length of the track 2 data matches the length specified in the ll parameter.
- There are two commands needed to program a mask. They are broken up into two parts to make the command shorter. Both commands need to be entered in to properly program a mask.
- The track 2 mask can specify a substring to extract. In this case, p1 and p2 specify the starting positions of the substrings, and ll1,ll2 specify the number of characters to extract. If ll1,ll2 are 000 then the remainder of the track 2 data (starting at the p1,p2 position) is extracted.
- If p1,p2 is 00, the ll1 or ll2 refer to an ascii chararacter which is inserted at the current position. Ascii characters

are entered in using their 3 digit decimal value.

Track 2 Mask #2 Command - Part 2

X068 p3 ll3 p4 ll4 p5 ll5

p3 = posistion 1
ll3 = length/constant 1
p4 = posistion 2
ll4 = length/constant 2
p5 = posistion 2
ll5 = length/constant 2

parameters same as above

notes

- There are 2 possible masks for track 2, which are executed if the length of the track 2 data matches the length specified in the ll parameter.
- There are two command needed to program a mask. They are broken up into two parts to make the command shorter. Both commands need to be entered in to properly program a mask.
- The track 2 mask can specify a substring to extract. In this case, p3,p4, and p5 specify the starting positions of the substrings, and ll3,ll4, and ll5 specify the number of characters to extract. If ll3,ll4,ll5 are 000 then the remainder of the track 2 data (starting at the p3,p4, or p5 position) is extracted.
- If p3,p4, or p5 is 00, then ll3,ll4,ll5 refer to an ascii chararacter which is inserted at the current position. Ascii characters are entered in using their 3 digit decimal value.

Proximity Commands

There are a number of commands to program the reading of proximity style rf cards. These commands apply to all rf cards read by the STAx series.

The reader can uniquely process up to three different card formats, including formatting and masking of the input data. In the "passthru" mode, there is no limit to the card formats accepted. The passthru mode is limited to 200 bits.

Proximity Input General Setup Command

X073 n m b p s

n	=	input mode	must be "1"
m	=	masking	0=off, 1=on
b	=	good read beep	0=off, 1=on
p	=	parity check	0=off, 1=on (26&28 bit cards only)
s	=	passthru	0=off, 1=on

notes

- Set "m" to a 1 if you want to perform masking on the decoded data.
- Set "b" to a 1 for a good read beep when the card is read.
- Set "p" to a 1 to turn on parity checking for 26 bit and 28 bit cards only.
- Set "s" to a 1 to pass thru proximity data undisturbed as bit characters.

Prox Format # 1 Settings, Part 1

X074 bbb sss sl b bss bl y

bbb	=	total bits	
sss	=	site (facility) start	
sl	=	site length	
b	=	site is lsb	0=no, 1=yes
bss	=	badge (id) start	
bl	=	badge length	
y	=	badge is lsb	0=no, 1=yes

notes

- Set "bbb" to the total number of bits on the card for Format # 1. Set this to 00 if no site code is used.
- Set "sss" to the starting bit position of the site code. Note the first bit is considered bit # 1, not 0.
- Set "sl" to the number of bits included in the site field.
- Set "b" to a 1 if the bits are lsb (backwards) within the site code field.
- Set "bss" to the starting bit position of the id field.
- Set "bl" to the number of bits included in the id field.
- Set "y" to a 1 if the id field is lsb.

Prox Format # 1 Settings, Part 2

X075 iss il b sc ic uc

iss	=	issue code start	
il	=	issue length	
b	=	issue code is lsb	0=no, 1=yes
sc	=	# of site characters	
ic	=	# of id characters	
uc	=	# of issue characters	

notes

- Set "iss" to the starting bit position of the issue code. Set this to 00 if no issue code is used.
- Set "il" to the number of bits included in the issue code field.
- Set "b" to a 1 if the bits are lsb (backwards) within the issue code field.
- Set "sc" to the number of characters to encode the site into. Set to 00 to ignore the site.
- Set "ic" to the number of characters to encode the id into. Set to 00 to ignore the ID.
- Set "uc" to the number of characters to encode theissue code into.. Set to 00 to ignore the issue code.

Prox Format # 2 Settings, Part 1

X076 bbb sss sl b bss bl y

bbb	=	total bits	
sss	=	site (facility) start	
sl	=	site length	
b	=	site is lsb	0=no, 1=yes
bss	=	badge (id) start	
bl	=	badge length	
y	=	badge is lsb	0=no, 1=yes

notes

- Set "bbb" to the total number of bits on the card for Format # 2. Set this to 00 if no site code is used.
- Set "sss" to the starting bit position of the site code. Note the first bit is considered bit # 1, not 0.
- Set "sl" to the number of bits included in the site field.
- Set "b" to a 1 if the bits are lsb (backwards) within the site code field.
- Set "bss" to the starting bit position of the id field.
- Set "bl" to the number of bits included in the id field.
- Set "y" to a 1 if the id field is lsb.

Prox Format # 2 Settings, Part 2

X077 iss il b sc ic uc

iss	=	issue code start	
il	=	issue length	
b	=	issue code is lsb	0=no, 1=yes
sc	=	# of site characters	
ic	=	# of id characters	
uc	=	# of issue characters	

notes

- Set "iss" to the starting bit position of the issue code. Set this to 00 if no issue code is used.
- Set "il" to the number of bits included in the issue code field.
- Set "b" to a 1 if the bits are lsb (backwards) within the issue code field.
- Set "sc" to the number of characters to encode the site into. Set to 00 to ignore the site.
- Set "ic" to the number of characters to encode the id into. Set to 00 to ignore the ID.
- Set "uc" to the number of characters to encode theissue code into.. Set to 00 to ignore the issue code.

Prox Format # 3 Settings, Part 1

X078 bbb sss sl b bss bl y

bbb	=	total bits	
sss	=	site (facility) start	
sl	=	site length	
b	=	site is lsb	0=no, 1=yes
bss	=	badge (id) start	
bl	=	badge length	

y = badge is lsb 0=no, 1=yes

notes

- Set "bbb" to the total number of bits on the card for Format # 3. Set this to 00 if no site code is used.
- Set "sss" to the starting bit position of the site code. Note the first bit is considered bit # 1, not 0.
- Set "sl" to the number of bits included in the site field.
- Set "b" to a 1 if the bits are lsb (backwards) within the site code field.
- Set "bss" to the starting bit position of the id field.
- Set "bl" to the number of bits included in the id field.
- Set "y" to a 1 if the id field is lsb.

Prox Format # 3 Settings, Part 2

X079 iss il b sc ic uc

iss = issue code start
il = issue length
b = issue code is lsb 0=no, 1=yes
sc = # of site characters
ic = # of id characters
uc = # of issue characters

notes

- Set "iss" to the starting bit position of the issue code. Set this to 00 if no issue code is used.
- Set "il" to the number of bits included in the issue code field.
- Set "b" to a 1 if the bits are lsb (backwards) within the issue code field.
- Set "sc" to the number of characters to encode the site into. Set to 00 to ignore the site.
- Set "ic" to the number of characters to encode the id into. Set to 00 to ignore the ID.
- Set "uc" to the number of characters to encode theissue code into.. Set to 00 to ignore the issue code.

Prox Format # 1 Mask Command - Part 1

X080 ll p1 ll1 p2 ll2

ll = length length to match, 00 = any length
p1 = posistion 1 start position, 00 means ll1 is a constant
ll1 = length/constant 1 length,000=take remaining, or constant value
p2 = posistion 2
ll2 = length/constant 2

notes

- There is a separate mask for each of the three programmable input formats.
- There are two commands needed to program a mask. They are broken up into two parts to make the command shorter. Both commands need to be entered in to properly program a mask.
- The mask can specify a substring to extract. In this case, p1 and p2 specify the starting positions of the substrings, and ll1,ll2 specify the number of characters to extract. If ll1,ll2 are 000 then the remainder of the 3 data (starting at the p1,p2 position) is extracted.
- If p1,p2 is 00, the ll1 or ll2 refer to an ascii chararacter which is inserted at the current position. Ascii characters are entered in using their 3 digit decimal value.

Prox Format # 1 Mask Command - Part 2

X081 p3 ll3 p4 ll4 p5 ll5

p3	= posistion 1	parameters same as above
ll3	= length/constant 1	
p4	= posistion 2	
ll4	= length/constant 2	
p5	= posistion 2	
ll5	= length/constant 2	

notes

- There are two command needed to program a mask. They are broken up into two parts to make the command shorter. Both commands need to be entered in to properly program a mask.
- The mask can specify a substring to extract. In this case, p3,p4, and p5 specify the starting positions of the sub-strings, and ll3,ll4, and ll5 specify the number of characters to extract. If ll3,ll4,ll5 are 000 then the remainder of the data (starting at the p3,p4, or p5 position) is extracted.
- If p3,p4, or p5 is 00, then ll3,ll4,ll5 refer to an ascii chararacter which is inserted at the current position. Ascii characters are entered in using their 3 digit decimal value.Prox Format # 1 Mask Command - Part 1

Prox Format # 2 Mask Command - Part 1

X082 ll p1 ll1 p2 ll2

ll	= length	length to match, 00 = any length
p1	= posistion 1	start position, 00 means ll1 is a constant
ll1	= length/constant 1	length,000=take remaining, or constant value
p2	= posistion 2	
ll2	= length/constant 2	

notes

- There is a separate mask for each of the three programmable input formats.
- There are two commands needed to program a mask. They are broken up into two parts to make the command shorter. Both commands need to be entered in to properly program a mask.
- The mask can specify a substring to extract. In this case, p1 and p2 specify the starting positions of the substrings, and ll1,ll2 specify the number of characters to extract. If ll1,ll2 are 000 then the remainder of the 3 data (starting at the p1,p2 position) is extracted.
- If p1,p2 is 00, the ll1 or ll2 refer to an ascii chararacter which is inserted at the current position. Ascii characters are entered in using their 3 digit decimal value.

Prox Format # 2 Mask Command - Part 2

X083 p3 ll3 p4 ll4 p5 ll5

p3 = posistion 1
ll3 = length/constant 1
p4 = posistion 2
ll4 = length/constant 2
p5 = posistion 2
ll5 = length/constant 2

parameters same as above

notes

- There are two command needed to program a mask. They are broken up into two parts to make the command shorter. Both commands need to be entered in to properly program a mask.
- The mask can specify a substring to extract. In this case, p3,p4, and p5 specify the starting positions of the substrings, and ll3,ll4, and ll5 specify the number of characters to extract. If ll3,ll4,ll5 are 000 then the remainder of the data (starting at the p3,p4, or p5 position) is extracted.
- If p3,p4, or p5 is 00, then ll3,ll4,ll5 refer to an ascii chararacter which is inserted at the current position. Ascii characters are entered in using their 3 digit decimal value.Prox Format # 1 Mask Command - Part 1

Prox Format # 3 Mask Command - Part 2

X084 ll p1 ll1 p2 ll2

ll = length
p1 = posistion 1
ll1 = length/constant 1
p2 = posistion 2
ll2 = length/constant 2

length to match, 00 = any length
start position, 00 means ll1 is a constant
length,000=take remaining, or constant value

notes

- There is a separate mask for each of the three programmable input formats.
- There are two commands needed to program a mask. They are broken up into two parts to make the command shorter. Both commands need to be entered in to properly program a mask.
- The mask can specify a substring to extract. In this case, p1 and p2 specify the starting positions of the substrings, and ll1,ll2 specify the number of characters to extract. If ll1,ll2 are 000 then the remainder of the 3 data (starting at the p1,p2 position) is extracted.
- If p1,p2 is 00, the ll1 or ll2 refer to an ascii chararacter which is inserted at the current position. Ascii characters are entered in using their 3 digit decimal value.

Prox Format # 3 Mask Command - Part 2

X085 p3 ll3 p4 ll4 p5 ll5

p3 = posistion 1
ll3 = length/constant 1
p4 = posistion 2
ll4 = length/constant 2
p5 = posistion 2
ll5 = length/constant 2

parameters same as above

notes

- There are two command needed to program a mask. They are broken up into two parts to make the command shorter. Both commands need to be entered in to properly program a mask.
- The mask can specify a substring to extract. In this case, p3,p4, and p5 specify the starting positions of the substrings, and ll3,ll4, and ll5 specify the number of characters to extract. If ll3,ll4,ll5 are 000 then the remainder of the data (starting at the p3,p4, or p5 position) is extracted.
- If p3,p4, or p5 is 00, then ll3,ll4,ll5 refer to an ascii chararacter which is inserted at the current position. Ascii characters are entered in using their 3 digit decimal value.Prox Format # 1 Mask Command - Part 1

Proximity Card Selection Options

X123 1 2 3 4 5 6 7 8

1 =	read hid cards	1=yes 0=no
2 =	read awid cards (26 bit)	1=yes 0=no
3 =	read farpointe cards	1=yes 0=no
4 =	unused	1=yes 0=no
5 =	read em cards	1=yes 0=no
6 =	read casi proxlite cards	1=yes 0=no
7 =	maxsecure required	1=yes 0=no
8 =	read IBC cards	1=yes 0=no

X127 1 2 3 4 5 6 7 8

1 =	rfu	must be 0
2 =	rfu	must be 0
3 =	read hid c&d cards	1=yes 0=no
4=	rfu	must be 0
5=	rfu	must be 0
6=	rfu	must be 0
7=	rfu	must be 0
8=	rfu	must be 0

notes

- Hid c&d cards require a card format setup of 64 bits with the id number starting t bit 1
- Awid support is for 26 bit cards only
- Casi Proxlite cards are decoded into a 12 digit number

Maxsecure Number

X124 xxxx

xxxx = 4 character hexidecimal value to match for maxsecure cards.

EM Card Options

X126 x

x =	# bits	0 = read all 40 bits, 1 = read 32 bits only

Speaker Programming Commands

Speaker Setup Command (firmware after 09/03/09)

X095 s fff gg

s =	startup beep	0=disabled, 1=enabled
fff =	frequency	001-255, default is 19
gg =	good read beep time	10ms increments (10ms thru 990ms)

notes

- With this command you can specify whatever frequency you like for the speaker. As the number gets higher, the frequency gets higher.
- Because the speaker is a pizzo, certain frequencies will appear to be louder than others. The loudest frequency setting is 19, which is the default.

Speaker Duty Cycle

X102 oo ff

oo =	on cyles	01-15
ff =	off cycles	01-15

notes

- To conserve power, the speaker can run less than 100% duty cycle. The defaults are on=01, off=03, or 25% duty cycle. Note that changing the duty cycle will also affect the perceived frequency.

Invalid Card Beep

X129 x

x = 0 off
x = 1 on

notes

- If x is set to 1, an invalid card beep will be heard whenever a card is read which does not match the programmed parameters (wiegand formats, etc..)

Led Programming Commands

General Led Settings

X011 s t b gg s t b gg

s = must be 0
t = startup — 0=off, 1=on, 2=red(bi) 3=green(bi) (2&3 refer to led1 only)
b = must be 0
gg= good read time

notes

- The above command pertains to JX mode only, and has no effect when in STAx mode.
- For inernally controlled leds only, the state of the led(s) on startup can be off or on. For the left led which may be a bicolor led, it can also be ON red or ON green.
- Internally controlled leds can also be set to blink slowly or fast on startup and remain in that state.
- gg specifies the number of seconds the led should go ON for when a "good read" occurs

Led Good Read settings

X012 a b

a = action led 1 — 0=on 1=off 2=blink on 3=blink off

b = action led 2 — 0=on 1=off 2=blink on 3=blink off

notes

- The above command pertains to JX mode only, and has no effect when in STAx mode.
- Either led 1 or led 2 can be turned on or blinked after a good read.
- The leds are turned on or blinked for the amount of time specified in the gg parameter of the X011 command.

Relay Commands

Relay Setup

X032 a n l tt

a =	autotrigger	0=off, 1=on
n =	startup position	0=off, 1=on
l =	must be 0	0=off, 1=on
tt =	autotrigger time	time in seconds, max=99

notes

- The above command pertains to JX mode only, and has no effect when in STAx mode.
- If autotrigger is on, then the relay is turned on for the autotrigger time whenever a good read occurs
- The relay position, when the reader is powered up (or restarted) can be programmed to be on or off.

General Programming Commands

Restart Command

X0151

no parameters

notes

- This command causes the reader to restart.

Transmit time delay for multitech readers

X038 tt

tt = wait and delay time 00-99 (*10ms)

notes

- When using combination readers that may have proximity and magstripe or barcode on the same card, this command allows you to program a wait time that the reader will wait after transmitting prox, barcode, or mag data, before processing one of the other inputs. This is designed to put a delay of time between two successive outputs.

Full Response for Commands

X107 t

t = full response 0=off, 1=on

notes

- STAx readers can emulate STA series readers, where no responose is returned when sending commands. If the full response setting is on, the the STAx reader will always respond with OK or NOK when a command is sent to it. Note that some commands are loosely checked by the reader, so the receipt of an OK does not in itself mean the command format was ok,. It merely means the reader received and processed the command.

Enable/Disable RF processing

X108 t

t =	process rf cards	0=off, 1=on

notes

- Rf card reading can be turned off with this command. This is useful when installing dual or tri technology readers; however you do not want to process rf cards at this time.

Process Cards Meeting Mask Requirements only

X109 x y z

x =	barcode	0=off, 1=on
y =	mag track 1	0=off, 1=on
z =	mag track 2	0=off, 1=on

notes

- If these options are on, then cards which are read, that do not match the length requirements for one of the programmed masks, are ignored.

Serial Port Commands

Serial Port General Setup

X014 b p ttt yyy1

b =	baudrate	1=1200 2=2400 3=4800 4=9600 5-19200 6=38400 7=57600 8=115200
p =	parity	1=8 none 2=7 even 3=7 odd
ttt =	terminator 1	001-127
yyy=	terminator 2	001-127, 000=none

notes

- The terminator character can be any ascii character from 001 thru 127. The terminator is entered in using the 3-digit decimal value of the ascii character, i.e. a carriage return is 013. You must have at least one terminator character.. If you are not using a second terminator character, set yyy to 000.
- Changing the terminator character(s) affects output only. Serial input is always terminated with a 0d hex.
- You cannot change the serial parameters for readers with tcp. The serial parameters are fixed for tcp units.
- Changes made to the serial port do not take effect until the reader is power cycled, or a restart command is issued.
- This command has no effect for TCP readers.

Serial Control Commands

The following commands are run-time commands which can be sent to the STAx reader when running in JX emulation mode. Most of these commands are the same as in the J series readers.

!xx Turn Relay ON for xx seconds, maximum 15
* Turn Relay OFF immediately
&xx Turn Beeper ON for xx*10 milliseconds, maximum 490ms
}xx Turn Red led (led 1) ON for xx seconds then turn off
xx=00=leave on permanently
{xx Turn Red led (led 1) OFF for xx seconds, then turn on
xx=00=leave off permanently
]xx Turn Green led (led 2) ON for xx seconds, then turn off
xx=00=leave on permanently
[xx Turn Green led (led 2) OFF for xx seconds, then turn on
xx=00=leave off permenently
)xx Blink Red led (led 1) for xx seconds, then leave on
(xx Blink Red led (led 1) for xx seconds, then leave off
>xx Blink Green led (led 2) for xx seconds, then leave on
<xx Blink Green led (led 2) for xx seconds, then leave off
? Poll command, for multidrop mode
V Return version identification string
Dxxxxxxxx Display message xxxxxxxx on alphanumeric display
Fx Rotate display, 0=normal(horizontal), 1=rotated(vertical)

Stand Alone Commands

Clear Schedules

This command clears all schedules in the reader. The format is:

SC

This command will return "A" if successful, nothing or "NOK" if not successful.

Add Schedule

This command adds a schedule into the reader, containing the allowable start and stop times for any particular day. The format is:

SAsssdhhmmhhmm	where: sss	=	schedule # (001 thru 099)
	d	=	day of week (1=monday,7=sunday)
	hhmm	=	start hours/minutes (0000-2359)
	hhmm	=	end hours/minutes

Schedule times are inclusive. If start hours/minutes = end hours/minutes, the schedule is active for 1 minute.

You cannot cross midnight with a schedule. If you need to cross midnight, then two schedules must be added and associated.

You cannot overwrite a schedule. Once the schedules in a reader are cleared, each schedule can be added only once. If you need to change the times for one specific schedule only, then all schedules need to be cleared, and then the new schedules downloaded.

This command will return one of the following:

A - schedule added
S - invalid schedule #
T - invalid time

Transmit Schedule

This command returns the current setting of a schedule, for all seven days. The format of the command is:

SRsss where: sss = schedule # (001-099)

The reader will then return a 56 byte stream in the format hhmmhhmm...hhmmhhmm, each hhmmhhmm being the start and end times for the schedule, for days 1 thru 7.

Associate Schedule

This command allows you to associate one schedule in tandem with another. Once you associate one schedule with another, the reader will first check the initial schedule for the employee. If the time does not fall within the initial schedule, then the associated schedule is checked, and any other schedules associated with the 2nd schedule. The checking does not stop until it reaches a schedule that has no associations.

For example, let's say you have a shift that runs from 1000 through 1800, but you do not want access between the hours of 1300 and 1400. The end result is that you will need access for the employee from 1000 through 1259, and from 1401 through 1800, on the same day. You cannot do this with one schedule but you can with 2. To do this with the two schedules - set up the first schedule (let's say schedule 001) with the times 1000 through 1259. Then, set up schedule 2 with the times 1401 through 1800. Then, send the Associate Schedule command to associate schedule 002 to schedule 001.

What will happen when you scan a card (assume the time is 1430) - is the reader will check schedule 1 which will fail. It will then check schedule 2 which will pass. The command is:

SSxxxyyy where: xxx = original schedule (001 through 099)
yyy = additional associated schedule (000 - 099)

Example:

SS001002 associates schedule 2 with schedule 1, meaning that any employee with schedule 1 will also check schedule 2 if schedule 1 fails.

Note that yyy must be greater than xxx. You can associate schedules only with schedules that have a larger schedule number. You can send the command SS001002 which means to associate 2 with 1, but you cannot send the command SS002001. This restriction is put in place so that you cannot create a situation where 2 schedules point to each other.

This command will return one of the following:

A - schedule associated
S - invalid schedule #
nothing or NOK if bad

Transmit Associated Schedule

This command will transmit to your computer the schedule association for a particular schedule.

STxxx where: xxx = schedule to transmit association for (001 through 099)

Example for schedule 001 - If schedule 001 has schedule 002 associated to it, this command will return 002. If it has no association, it will return 000.

Add ID into List

This command adds an ID into the reader, along with a schedule # for the ID.

To enter in an ID with no schedule information, use a schedule number of 000.

The format of the Add ID command is:

ACsssiiiiiiiiii...	where: sss	=	schedule # (000 thru 099)
ANsssiiiiiiiiii...	iiii...	=	ID number

Either command (AC or AN) can be used to add an ID. The STAx reader does not perform a check for duplicate entries when the AC command is used. AC is for compatibility purposes only.

The possible return values for this command are:

A-id added
N-id not added
S-invalid schedule #

Reset Record Size and Clear Memory

This command resets the reader, clears the memory in the reader, and sets the id size for employee id's in the reader. The minimum allowable id size is 4, and the maximum is 40. By default, the reader id size is set to 18. The format of this command is:

\ss	where: ss	=	id size (4-40)

This command only needs to be used once, to set up your reader, to a record size other than the default 18.

Set Mode

This command sets the operating mode of the reader - SA, STA, or "JX" emulation mode. SA readers can be set to either SA mode or "J" emulation mode. They cannot be set to STA mode. STA readers can be set to any of the modes. Effectively, setting an STA reader to SA mode really does nothing, except turn off the transactional logging.

The format of this command is:

|m where: m = 1 ("JX" emulation mode)
3 (SA mode)
4 (STA mode)
5 (Accumulate mode)

Please note that "|" is a vertical bar, not a lower case L.

Set Time

This command sets the time in the time clock in the reader. The format of the command is:

+yymmddhhmmssw where:

yy	=	year
mm	=	month
dd	=	day
hh	=	hour
mm	=	minute
ss	=	seconds
w	=	day of week (1=monday,7=sunday)

You should always reset the time in the reader after getting the reader from the factory, because the time is not always set at the factory prior to shipment.

This command returns the following:

T-bad time
O-time set

Set Relay/Green Led Time

This command sets the amount of time the relay is on for, and also the amount of time the green led blinks for, when access is granted. The maximum amout of time is 99 seconds, and the minimum is 1 second.

The format of this command is:

"ss where:ss = number of seconds (01-99)

The default in the reader is 5 seconds.

Set Bad ID Red Led Blinking

Whenever a bad id is scanned (or a good id which does not match the time restrictions), the red led blinks for a predetermined number of seconds. This command allows you to set the number of seconds, from 1 to 15. The format of the command is:

`ss where:ss = number of seconds (01-15)

The default in the reader is 5 seconds.

Set Bad ID Logging on/off

Whenever a bad id is scanned (or a good id which does not match the time restrictions), an entry can be made in the transactional logging. This command allows you to change the status of the logging to control whether bad id scans are saved in the log, or only good scans (for which access was granted), are saved in the log. The format of the command is:

:x where:x = 0 - bad id logging off
1 - bad id logging on

The default in the reader is bad logging on.

Return Number of Records

This command returns the number of total available records in the reader for the access control list, and for the transactional log. Also, the total number of records used in the access list and the log are returned. Use this command to see if the reader is becomming full and an upload of the log may be required.

The format of this command is:

z

The command will return:

ttttddddddxxxxxllllll where:

ttttt	=	total # available id's in the access list
ddddd	=	total # access list records used
xxxxx	=	total # available log entries
llllll	=	total # log entries used

Read Time

This command returns the time which is stored in the reader. The format of the command is:

t

Clear Log

This command clears all of the log (transaction) entries in the reader, and is usually issued after you have completed uploading all of the transactional data. The format of the command is:

cl

Clear Downloaded List

This command clears all of the employee id's in the reader and is usually done prior to a complete download of the reader. The format of the command is:

cd

If the reader is set to the "active" memory bank, all active ID numbers is the reader are deleted. If the reader is set to the "background" bank, then the background memory bank is cleared, which does not affect the current active list in the reader. To clear the background bank, send the BACK command then send the cd command.

Upload Transactional Log Item

This command uploads one transaction from the Transactional Log in the reader. Each time this command is sent to a reader - the next transaction is returned. After all transactions have been uploaded, this command returns "END".

The format of the command is:

l (lowercase L)

The command returns:

sYYMMDDHHMMSSiiiiiiiiiiii...

where s is the status
YYMMDDHHMMSS is the date/time
iiiiiiii.... is the id

or END

The status which is returned can be any of the following:

0=transaction ok (access granted)
1=bad employee id
2=employee id ok, but not within scheduled time (outside of time setup in schedule)

After all items have been uploaded, this command returns END.

The STAx uses a circular buffer. Therefore, when you issues this command you will always get the oldest log item that has not already been uploaded.

Log Reset

You can reset the log pointer so that the l command will give you the oldest transaction still in the reader, not the oldest which has not been updated.

The format of the command is:

lr

Set to Background For Downloading Command

This command tells the reader that any download of ID numbers are to go to the background memory bank and not to the currently used bank. This command is typically used prior to performing a full download of ID's into the reader, into the background bank, so the reader remains fully functional during the download.

Note that this command will also affect the ID record count data which is returned with the z command, since you will now receive that information which pertains to the background bank. The log counts are not affected.

The format of the command is:

BACK

Set to Current Bank For Downloading Command

This command tells the reader to move back to the currently operational bank. Any new add commands will add data to the current operational bank.

The format of the command is:

END

Swap Banks

This command makes the background bank the currently active bank, basically logically replacing the data in the current operational bank with the data in the background bank.

Typically, for a full system download of ID numbers, you would first issue the BACK command to set the system to the background bank for downloading, then download the records, then SWAP the bank back.

The format of the command is:

SWAP